



青岛大学
QINGDAO UNIVERSITY

本科毕业设计

题 目 智能交通隔离带系统——LoRa 协调器的设计与实现
学 院 计算机科学技术学院
专 业 物联网工程
学生姓名 王璐
指导教师 范延滨

2021 年 6 月

智能交通隔离带系统—LoRa 协调器的设计与实现

Design and implementation of LoRa Coordinator for Intelligent
Traffic isolation system

郑重声明

本人呈交的学位论文（设计），是在指导教师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。除文中已经注明引用的内容外，本学位论文（设计）的研究成果不包含他人享有著作权的内容。对本论文（设计）所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文（设计）的知识产权归属于青岛大学。

本人签名： 王璐

日期： 2021年5月16日

摘要

物联网是基于因特网和传感器网络的新兴技术，物联网技术的发展带动了智慧城市、智慧交通领域的飞速发展，其是“互联网+”的重要功能拓展，对于构建新型城市、实现基础设施的智能化管理起到了至关重要的作用。

本文分析了智能交通隔离带系统对于网关的需求，并基于 MQTT (Message Queuing Telemetry Transport) 协议、LoRa (Long Range) 协议和嵌入式 STM32 系统，针对智慧交通的应用场景，完成了 LoRa 自组网网关的设计与实现。该 LoRa 网关实现了 LoRa 结点入网管理、数据报向上/向下转发、必要信息的显示等功能。

本网关的设计方案既包括硬件电路也包含软件架构。硬件上，基于“先分块-后整合”的设计方案，将 MCU 与 LoRa 模块、电源模块、USB 转串口模块、Wifi/4G 模块和显示模块等相连接并共同置于一块 PCB 上；软件上，网关的软件设计采用分层的思想，从上至下分别是应用层、中间件层、操作系统层、板级支持包和驱动层。其中，驱动层将对 STM32 内部外设的配置和功能函数进行封装；板级支持包负责将硬件模块的初始化、寄存器读写等函数封装供上层调用；操作系统层实现对 FreeRTOS 系统分配堆栈空间，设置队列最大长度、任务最高优先级等基本参数，初始化所需的硬件资源；中间件层是将硬件与现实应用相结合的关键，它负责实现所需的功能函数；应用层针对智慧交通隔离带的应用场景需求，完成业务逻辑的实现。

关键词 物联网 智慧交通 嵌入式 网关 LoRa MQTT STM32

Abstract

The Internet of things is a new technology based on the Internet and sensor networks. the development of it has led to the rapid development of smart cities and intelligent transportation, which is an important functional expansion of "Internet +". It plays a vital role in building a new type of urban infrastructure, intelligent management and services.

This paper analyzes the requirements of intelligent traffic isolation system for gateway, and based on MQTT (Message Queuing Telemetry Transport) protocol, LoRa (Long Range) protocol and embedded STM32 system, completes the design and implementation of LoRa gateway system for the application scenario of intelligent transportation.

The design of this gateway includes both hardware circuit and software architecture. In hardware, based on the design scheme of "block first-then integration", MCU and LoRa module, power module, USB to serial port module, Wifi/4G module and display module are connected and placed on a PCB together.. In software, the gateway adopts a layered design method, which includes application layer, middleware layer, operating system layer, board-level support package and driver layer from top to bottom. The driver layer encapsulates the configuration and function of STM32's internal peripherals; The board-level support package is responsible for the basic operation of each hardware module and completes the preparation before entering the operating system; The operating system layer implements the initial configuration of FreeRTOS and allocates the required hardware resources; The middleware layer realizes the function called by the application layer by calling each other among different modules; The application layer is responsible for the implementation of the business logic according to the application scenario requirements of the intelligent traffic isolation zone.

Keywords Internet of things Intelligent transportation Embedded system Gateway
LoRa MQTT STM32

目 录

摘要	I
Abstract	II
1 绪论	1
1.1 研究背景	1
1.2 国内外研究现状	1
1.2.1 智慧交通研究现状	1
1.2.2 物联网网关研究现状	2
1.3 论文组织结构	3
2 相关技术介绍	5
2.1 LoRa 远距离通信技术	5
2.1.1 LoRa 概述	5
2.1.2 LoRa 数据包结构和数据格式	6
2.2 MQTT 消息传输协议	6
2.3 JSON 数据传输格式	8
2.4 FreeRTOS 嵌入式实时操作系统	9
2.5 本章小结	9
3 网关需求分析	11
3.1 物联网网关系统需求分析	11
3.1.1 功能性需求	11
3.1.2 非功能性需求	11
3.2 本章小结	12
4 网关系统硬件的详细设计	13
4.1 网关的硬件总体结构设计	13
4.2 网关主要元件选型	13
4.2.1 MCU 选型	13
4.2.2 LoRa 模块选型	14
4.2.3 WiFi/4G 模块选型	16
4.2.4 显示模块选型	17
4.2.5 串口调试模块选型	18
4.2.6 电源模块选型	19
4.3 电路板布线及抗干扰设计	20
4.4 本章小结	21

5	网关系统软件的详细设计	22
5.1	网关系统的软件框架	22
5.2	驱动层	22
5.3	板级支持包	23
5.3.1	LoRa 模块初始化	23
5.3.2	WiFi 模块初始化	24
5.3.3	显示模块初始化	25
5.4	操作系统层	26
5.4.1	STM32CubeMX 配置 FreeRTOS	26
5.4.2	初始化硬件模块	28
5.5	中间件层	28
5.5.1	连接 MQTT 服务端	28
5.5.2	MQTT 的订阅和发布	29
5.5.3	LoRa 的发送和接收	30
5.5.4	解析 JSON 数据	31
5.5.5	中断服务函数实现	32
5.6	应用层	34
5.6.1	MQTT 数据收发任务	34
5.6.2	LoRa 数据收发任务	35
5.6.3	OLED 显示任务	36
5.6.4	事件标志组	36
6	总结与展望	38
6.1	总结	38
6.2	展望	38
	致谢	40
	参考文献	42

1 绪论

1.1 研究背景

美国麻省理工学院 Auto-ID 研究中心于 1999 年最早定义了物联网的概念^[1]。我国在 2010 年 3 月的政府工作报告^[2] 中对物联网有如下说明：（物联网）是指通过信息传感设备，按照约定的协议，将互联网与实实在在的物品连接起来，进行信息交换和通讯，以实现智能化跟踪管理的一种网络。它突破了原本对于网络的定义，简单来说，传统网络参与的主要主体是人，人们使用上网工具在网络中交换信息，而物联网则将原本独立的实物带入网络中^[3]，比如通过物联网实现智能抄表，利用物联网技术制造智能家居产品，实现远程遥控、智能启停等功能。伴随着越来越多新技术的发展与应用，“万物互联”已经成为人类社会不可阻挡的趋势，物联网也必然大有用武之地。

智慧城市是利用不断发展的信息技术和网络，对民众生活、出行、城市公共设施、安全等方面的需求给出智能化的解决方案。以便更好的符合可持续发展的理念。智慧交通是指利用互联网信息将汽车运行动态行程进行汇总，从而得到一种更高效、更便捷、更安全的交通运输方式，带动运输业的发展。而将智慧交通融入于智慧城市中，可以更好的带动城市的经济发展，都说运输是城市的核心，交通发展的好，城市才能得到更好的发展，好的交通服务，可以为人们服务，人民受益，对国家的好感度就会上升，也更加有利于国家政策的推行以及国家经济的发展^[4]。

1.2 国内外研究现状

智慧交通伴随着新兴科技的发展和普及正处于蓬勃发展的势头中，物联网技术是应用于智慧交通背景下的一项重要科技。物联网架构分为应用层、网络层和感知层，大部分人将网关划分为网络层中，其将整个感知网络 and 传统网络互联起来，使应用层避免实现复杂的协议转换，增加了整个物联网系统的效率。

1.2.1 智慧交通研究现状

智慧交通是依托物联网技术等现代电子科学技术，在原有交通运输体系基础上发展形

成的智能化服务系统，对交通管理、交通运输及交通基础设施等领域进行全方位管控，使得交通运输服务系统具有环境感知、实时报告、状况预测以及远程控制等智慧功能，通过优化交通资源配置结构，以缓解城市交通出行矛盾，充分发挥道路基础设施效能，为车辆、行人提供便利的出行环境。

智能交通隔离带作为智慧交通的重要组成部分，正受到越来越多的关注和重视。张若雯^[5]等人通过分析现有潮汐车道存在的问题，设计了一款伸缩式的可移动潮汐车道，通过远程控制方式根据车流状况组成不同的潮汐车道，对缓解主城区道路车流量不平衡，车道利用率低起到了重要作用。

1.2.2 物联网网关研究现状

物联网网关作为整个物联网架构的中心结点，在大多数基于网状拓扑、星型拓扑架构的物联网系统中不可或缺，网关的一大重要任务是实现不同种类的协议转换，包括传感网络-传感网络，也包括传感网络-传统 Internet。除此之外，物联网网关还必须实现对感知层终端结点连接状态的管理。

我国的传感器网络标准工作组也根据用途、接口类型、使用协议等方面对网关进行分类，指定相关接口、协议、使用场景等标准，向标准化、合理化的网关设计迈出新的一步。与此同时，国内从事物联网行业的相关企业和科研院所也致力于打造解决环境中组网问题的新型物联网网关^[6]。

庄旭菲^[7]等人结合物联网的技术特点和农业物联网的应用需求，提出了一种基于 Zigbee 和 HTTP 技术的嵌入式 Linux 农业物联网网关，该网关与多个农田传感终端连接，实现农田传感数据的实时上传，完成 Zigbee 与 Internet 的互联互通。然而，HTTP 协议并不适用于存储和计算性能较弱但数据流量大的物联网设备中。针对这一问题，徐侃^[9]等人设计了基于 MQTT 协议的网关设备，它使用 MQTT 协议与上层应用进行通信，实现简单，占用极少的硬件资源，非常适合现代物联网应用环境。论文研究内容

本文分析了智慧交通隔离带网络系统对于网关的需求，并基于嵌入式 STM32 处理器，使用 LoRa 协议、MQTT 协议，设计实现了新型物联网网关。该 LoRa 网关作为整个智慧交通隔离带网络的中心，联通了 LoRa 自组织网络 and 传统 Internet 网络，能并且实现了例如入网管理、报文转发、协议转换等功能。本文从硬件和软件两方面来实现对网关的设计与实现。

1.2.3 论文研究内容

论文主要研究内容如下：

1. 对智慧交通场景下的物联网网关进行需求分析，为之后对网关架构设计指明道路。
2. 网关的硬件采用模块化的设计思路，对用到的 LoRa 模块、WiFi 模块、MCU 等基于交通道路环境进行选型，使用先分离-后整合的方式，最终将所有模块置于一张 PCB 上。
3. 基于分层的软件设计架构，为所选的硬件模块设计合适的代码逻辑，使之处于良好的运行状态。本网关的软件设计主要从与上层和下层的交互出发，使用 LoRa 技术和 MQTT 协议来解决系统中传感信息上传和命令下发等功能。网关系统的软件设计分为五层，从上到下分别是应用层、中间件层、操作系统层、板级支持包和驱动层。

1.3 论文组织结构

本小节将介绍论文的整体组织架构安排，文章共分六个章节，每章的主旨大体如下：

第一章 描述了物联网、智慧交通领域的发展现状，引出论文的选题背景和设计本网关的重要意义。对论文的主要架构进行总结。

第二章 介绍网关设计中使用的协议和技术，包括 LoRa 协议、MQTT（消息队列遥测传输协议）、JSON 数据格式、FreeRTOS 嵌入式实时操作系统。

第三章 基于智慧交通隔离带的应用背景，对网关的需求进行详细分析。

第四章 根据对 LoRa 网关的需求分析，对智慧交通隔离带网关使用的硬件模块进行选型和 PCB 设计，使之满足交通道路应用场景下的需求。

第五章 根据对智慧交通隔离带网关的需求分析和硬件设计，分层次对网关的软件进行开发，实现协议转换、数据转发、接入管理等功能，以满足第三章总结的需求。

第六章 对论文主要工作进行总结，并展望物联网网关在智慧交通领域美好前景。

2 相关技术介绍

在整个智慧交通隔离带系统中，结点与网关之间的通信采用 LoRa 远距离通信协议。上行链路中，网关负责对接收的传感数据进行解析，使用 JSON 格式进行封装，再通过 MQTT 协议交付上层应用；下行链路中，网关将对服务器发来的 json 格式的字符串命令进行解析，并通过 LoRa 协议发送给相应的结点。网关部署 FreeRTOS 系统，用于实现任务调度、事件通知、内存安全分配与回收等功能。

本章将具体详细的介绍 LoRa 协议、MQTT 协议、JSON（JavaScript Object Notation）数据格式和 FreeRTOS 嵌入式实时操作系统。

2.1 LoRa 远距离通信技术

2.1.1 LoRa 概述

LoRa（Long Range，基于扩频技术的超远距离无线传输方案）是 IBM 公司和 Semtech 合作推出的 LPWAN 远距离物联网技术，主要使用 1GHz 以下非授权频段，中国 LoRa 使用频段有 470~510MHz 和 779~787MHz^[10]。LoRa 联盟是美国圣特公司于 2015 年组织成立的非盈利组织，到目前为止，LoRa 联盟的成员已遍布世界各地，具体分布如图 2-1 所示：

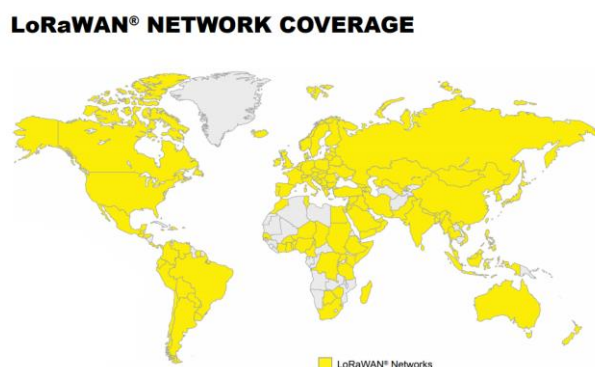


图 2-1 LoRa 联盟成员分布

LoRa 采用扩频、跳频技术实现超远距离的无线传输，并且通过“发送后休眠”、“CAD 定时检测”等机制实现了高效的能源节约。目前，与另一远距离通信技术 NB-IOT 相比，LoRa 的优势在于其在全球免费频段运行，包括 433、868、915MHz 等，用户无需付费申请频段^{[11][12]}。

2.1.2 LoRa 数据包结构和数据格式

如表 2-1 所示，LoRa 数据包由前导码（Preamble）、可选报头（Header）、数据有效负载（Payload）三部分组成。

表 2-1 LoRa 数据包结构

Preamble	Header	CRC	Payload	Payload
	(仅显式模式)			CRC

1. 前导码

LoRa 通信中接收方通过探测前导码来接收数据包，前导码的默认长度为 12，但实际上，前导码的是可扩展的，其范围在 6+4~65535+4 之间。接收端的前导码长度应与发送端保持一致，这样接收端才能检测到发送端要传输的数据。

2. 报头

显式报头模式会在 LoRa 数据包中携带本次的配置信息（有效负载长度、CRC 校验和编码率），每个数据包的配置不必都相同；而隐式报头模式则将配置信息写入 RegModemConfig1 寄存器中，这就表示它对于每个 LoRa 数据包都固定值不改变。

3. 有效负载

发送端要发送的数据就被存放在 LoRa 数据包中的 Payload 字段中。

2.2 MQTT 消息传输协议

随着云计算技术的快速发展和广泛应用，物联网系统借助云端实现对本地区终端的管理和控制的需求日益增加。为了实现感知层和应用层的数据交互，就需要一种高效、简单、可靠的应用层数据传输协议。

传统网络应用协议代表为 HTTP（Hyper Text Transfer Protocol，超文本传输）协议，而当 HTTP 协议应用在物联网环境中时，会出现诸多问题^[9]：

- 1. 协议开销大。HTTP 协议中客户端与服务端建立的连接仅在一次“请求和应答”中有效。而物联网应用中的通信通常不是一次发送大量数据，所以每次有新的数据传输都需重新建立 HTTP 连接而导致重大的开销。
- 2. 无法实现服务器向主要向网关发起数据通信，只能实现单一的向上通信。不适合需要从应用层下发命令的场景，具有较大的局限性。
- 3. 缺乏隐私管理。HTTP 协议的数据传输都是明文，在对安全要求较高的场景下，

HTTP 通信极易被抓取数据包做恶意攻击，从而造成巨大的财产损失甚至威胁生命安全。

1999 年 IBM 公司提出了一种应用于物联网场景下的轻量级数据通信协议，相比于其它传统网络应用层协议，它占用的资源非常少。2014 年，MQTT 发展到了 3.1.1 版本，并被国际标准接纳，成为广泛应用于终端设备连接互联网场景中的网络协议。

MQTT 基于传输层的 TCP 协议，能够保持客户端与服务端的长连接。不同客户端之间通过发布/订阅消息的模式实现数据交换，这种模式使数据传输更加简单和灵活，同时能够降低应用程序的耦合性，实现消息的异步收发。物联网下的设备通常性能较弱，网络带宽小，在这种应用场景下，MQTT 能够很好地发挥其轻量级的优点，在节省网络带宽和性能的基础上连接终端设备和云端服务器，满足越来越多物联网系统与 Internet 连接的需求。

具体而言，MQTT 拥有以下特点^[13]：

1. 实现多种路由机制的数据传输，能够将末端设备在时空间上解耦，实现消息的异步收发。
2. 协议的固有开销较小，同时支持三种服务质量，传输效率较高，能够很好地适用物联网这种资源受限的环境。
3. 实现难度小，能够很好地处理并发连接问题，因此支持大数量级的客户端接入。

MQTT 协议通过消息进行数据传输，消息是 MQTT 的基本传输单元，异于 HTTP 的请求/应答机制，MQTT 的消息是通过订阅/发布机制在网络间流动，如图 2-2 所示。



图 2-2 MQTT 发布订阅机制

MQTT 由三个基本组件构成：订阅者，发布者和代理服务器。订阅者和发布者都是 MQTT 客户端，一个客户端可以同时是发布者和订阅者。代理服务器也称为消息服务器，它是数据转发的中心，负责客户端的连接请求、处理网络中的订阅、发布请求、消息的转发，代理服务器通常是部署在云端的应用程序或一种实体设备。消息的过滤标准是主题，接收方通过向代理服务器发送指定报文来订阅主题，而代理服务器按照主题对消息进行筛选和转发，完成基本的传输过程。在代理服务器中，往往还实现了一些安全性功能，如用

户鉴权等^[14]。

在 MQTT 通信过程中，主要通过消息体的固定报文格式来进行信息类型的甄别，并根据控制报文的类型完成不同的通信操作^[15]。MQTT 消息报文主要由以下三个部分组成，如表 2-2 所示：

表 2-2 MQTT 报文结构

固定报头		可变报头		消息载荷
报文类型及标志位	剩余长度	标识 MSB	标识 LSB	

- ◆ 固定报头：由两个字节组成。第一个字节包含 4 位的 MQTT 控制报文类型，和 4 位的标志位。MQTT V3.1.1 版本共支持 15 种报文类型，标志位仅 PUBLISH 报文使用；第二个字节是整个 MQTT 报文减去固定报头的字节数。
- ◆ 可变报头：在某些特定的消息类型中会添加一个可变报头。例如不同服务质量的消息会通过该报头的标识符告诉通信双方，并执行不同的操作。
- ◆ 消息载荷：报文中的应用信息，也即通信双方需要发送、获取的有效数据。在许多 MQTT 的解决方案中，也会加入一些消息加密和安全认证技术，使得协议实现应用安全标准。

2.3 JSON 数据传输格式

JSON 是一种基于纯文本的数据交换格式，易于人们阅读和编写，各类语言大多都有相应的库来解析和组装 JSON 格式的字符串，故其适用于跨平台、跨语言的数据传输，且具有直观、简单、轻量化的特点。之前大多用于 JS 与 Web 应用中，而近些年在物联网应用中被广泛使用。

JSON 数据格式规则如表 2-3 所示

表 2-3 JSON 格式规则

符号（英文）	作用
大括号 {}	用来描述一组“不同类型的无序键值对集合”
方括号 []	用来描述一组“相同类型的有序数据集合”
冒号:	映射符；键名与键值之间通过冒号进行分割
逗号,	分隔符；分割集合中的多个子项

双引号""	定义符;为了便于不同语言解析 JSON 格式,键名都用双引号包含
-------	----------------------------------

本系统中网关和服务器之间使用 JSON 格式传递数据,能极大减少服务器端开发人员的工作量,更加方便与各类不同的服务器进行通信。

2.4 FreeRTOS 嵌入式实时操作系统

FreeRTOS 是一款开源、免费的,支持多种硬件架构的实时操作系统。FreeRTOS 占用空间极小,编译后的可执行文件大小小于 10KB,核心代码不超过 8000 行,便于不用硬件系统进行移植。就是在如此微型的系统中实现了嵌入式应用中最常用到的任务切换、消息通知、内存管理等功能。

FreeRTOS 的整个系统基于三大模块来实现:

- ◆ 任务:一个任务就是一段代码,代码中通常包含一个永久执行的循环。用户可以在创建任务时指定其优先级、分配堆栈大小等参数。
- ◆ 任务间通信:FreeRTOS 中使用事件标志组和队列来实现任务间通信,通过信号量也能完成任务的互斥和同步操作。通过任务间的通信可使不同任务组合实现更强大的功能逻辑,在项目中会经常使用任务间的通信。
- ◆ 硬件接口:FreeRTOS 是位于底层硬件之上,中间件之下的层次,它有一部分代码是需要调用板级支持包提供的接口来实现例如各硬件模块的初始化、内存分配与回收,时钟的配置等功能。

2.5 本章小结

本章对智慧交通隔离带系统的 LoRa 网关数据业务处理所涉及到的协议和规范进行了简单介绍。网关通过 LoRa 远程通讯技术连接多个传感终端结点,实现感知层的通信,本章总结了 LoRa 技术的概念、发展和 LoRa 数据包的结构;此外,本章还对网关系统与上层应用通信时使用的 MQTT 协议的特点和数据包格式、JSON 格式的使用规则进行了大致的描述。最后,基于应用场景对于实时性有一定要求,本网关使用 STM32CubeMX 工具移植了 FreeRTOS 系统,使得业务处理逻辑更加清晰,提升了内存管理、事件通知的效率和安全性。

3 网关需求分析

物联网网关是整个智慧交通隔离带系统的中心，它承担着传感结点接入管理、上下层协议转换，数据包解析等功能。本章将从功能性与安全性、可靠性、用户友好等非功能方面对网关的实际需求进行详细分析。

3.1 物联网网关系统需求分析

3.1.1 功能性需求

在整个智慧交通隔离带系统中，网关是作为连接 LoRa 网络与 Internet 网络的桥梁，故智慧交通隔离带网关的设计必须满足以下功能需求：

1. 具有简单的网络管理的功能。能够管理终端结点的接入和退出，实现以网关为中心的星型网络拓扑。
2. 具有协议转换的功能。LoRa 数据包的负载要能被提取并且封装为 MQTT 报文的负载，反之亦然。
3. 具有报文转发的功能。对于不同目的的报文能正确处理并转发到相应的网络中。
4. 具有一定的存储空间。方便存放设备信息、配置信息和已连接的终端结点信息等。
5. 直接多终端连接。能够允许超过 100 个结点接入网关，并实现大量数据的并发处理。

3.1.2 非功能性需求

在网关满足基本的功能的前提下，其还应具备某些特性以保证在突发意外或者恶劣条件情况下的正常工作。这些特性包含安全性、用户体验性、可靠性等，具体分析如下：

1. 具有一定的安全保护能力。采用一些数据校验、连接认证等算法，防止正常报文传输差错和他人的恶意报文攻击。
2. 具有与外界交互的功能。通过 OLED 显示屏和按键组合实现对网关的在线配置和显示特定的信息。
3. 具有一定的可靠性。本网关大多处于长期工作的环境下，因此整个网关平台需拥有

可长期工作稳定、故障率低和易于维护等特性。为保证以上特定，该网关需具有合理的软硬件设计，故障率低，完整的出错解决方案。

3.2 本章小结

本章对 LoRa 网关的设计进行了详细的需求分析。功能性需求方面，由于其作为整个网络系统架构的传输层，必须具备接入管理、协议转换、报文转发等功能。非功能需求方面，网关设备还必须具备一定的安全性、可靠性和用户友好性。

本章对于之后 LoRa 网关的硬件和软件设计提供了重要的参考依据，更加详细的软硬件设计将在后文一一介绍。

4 网系统硬件的详细设计

硬件是整个网关的基础，决定着整个智慧交通隔离带系统的实现性能和运行效果。根据上文对其整体需求的分析，本章将阐述应用于智慧交通中的物联网网关的硬件设计方案，对主要的功能元件进行选型，以及外围扩展模块的接口设计。

4.1 网关的硬件总体结构设计

网关硬件按照模块化的设计风格，主要包含处理器、LoRa 通讯模块、WiFi/4G 模块、串口调试模块、显示模块和供电模块，各模块的关系如图 4-1 所示（实线表示通讯关系，虚线表示供电关系）。

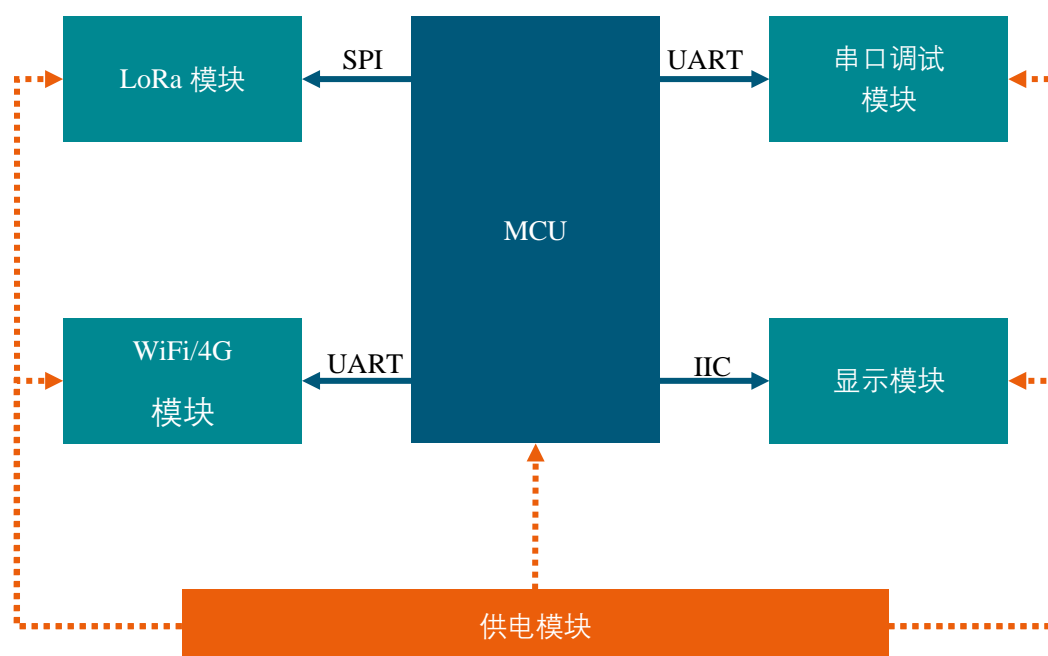


图 4-1 网关的硬件架构

4.2 网关主要元件选型

4.2.1 MCU 选型

MCU（Microcontroller Unit，微控制器）作为网关的核心元件其主要作用是运行编写好的程序代码，统筹调度各功能模块的运行，执行程序策略以及对各种事件进行处理。本网关采用的 MCU 型号为 ST（意法半导体）公司生产的 STM32F103RCT6。其具体参数如表

4-1 所示。利用 Keil 和 ST 官方提供的 HAL 库可以方便的开发 STM32 应用程序，并使用 SWD、JLINK 等进行便捷的 DEBUG 调试。同时，本项目利用 STM32CubeMX 开发工具对 STM32 进行快速的工程建立和外设配置，它还能进行第三方系统的配置，其中就包括本网关系统使用的 FreeRTOS 嵌入式实时操作系统。

表 4-1 STM32F103RCT6 的参数信息

外设	STM32F103RCT6	
Flash（KB）	256	
SRAM（KB）	48	
FSMC	No	
定时器	通用	4 个（TIM2、TIM3、TIM4、TIM5、）
	高级	2 个（TIM1、TIM8）
	基本	2 个（TIM6、TIM7）
通信接口	SPI	3 个（SPI1、SPI2、SPI3）
	IIC	2 个（IIC1、IIC2）
	USART	5 个（USART1~USART5）
	USB	1 个（USB2.0）
	CAN	1 个（2.0B 主动）
	SDIO	1 个
GPIO 端口	51 个	
12 位 ADC	3 个 ADC，每个 ADC16 个外部通道	
12 位 DAC	2 个	
工作电压	2.0V~3.6V	

4.2.2 LoRa 模块选型

网关通过 LoRa 远距离通信协议与终端结点通信，采用星型的拓扑结构，一个网关可连接超过 200 个终端节点。本网关采用 Ai-Thinker 公司的 Ra-02 LoRa 射频模块（如图 4-2 所示）。



图 4-2 Ra-02 LoRa 射频模块实物图

Ra-02 LoRa 射频模块将 SX1278 射频芯片和其外围电路进行封装，此模块下使用 LoRa™调制技术可实现最高+20dBm 的发射功率，传输举例空旷地可达 5km，市区可达 2~3km，可靠性高。接收状态下，其电流可达 20mA，休眠状态下，最低电流可达 10 μ A，最大成都减少了电源消耗，非常适合于电池供电的应用场景，LoRa™调制技术解决了传统无线传输技术无法同时满足长距离和低功耗的问题。LoRa 模块的主要参数如表 4-2 所示。

表 4-2 LoRa 模块主要参数

封装	SMD-16
尺寸	17*16*3.2(\pm 0.2)MM
天线形式	IPEX 座子
频谱范围	410MHz~525MHz
工作温度	-40 $^{\circ}$ C ~ 85 $^{\circ}$ C
供电范围	2.7~3.6V， 典型值 3.3V， 电流大于 200mA
支持接口	SPI
可编程比特率	最高达到 300kbps
认证	REACH、 RoHS

LoRa 模块与 MCU 通过 SPI 接口通讯，其支持三种方式的寄存器配置：

表 4-3 LoRa 模块的三种访问方式

单访问	仅执行一次的数据读或写操作，在发送地址字节之前将片选引脚置为低电平，之后发送地址字节、读写位，接着是读取的字节内容（若此次访问为读访问）。最后将片选引脚置为高电平
突发访问	执行多次的数据写入或者读取。在发送地址字节之前将片选引脚设为低电平，之后发送地址字节、读写位，在之后进行的若干次读写操作中地址是自动增加的，在最后一次字节写入或读取完成后将片选引脚置为高电平。
FIFO 访问	执行多次的数据写入 FIFO 或者从 FIFO 中读取。如果读或写操作的地址序列为 FIFO 的地址，那么该操作的数据将会被放入 FIFO（若该操作为写操作）或返回 FIFO 中的数据（若操作

	为读操作)，片选位在最后一个字节读写完成后被置为高电平。
--	------------------------------

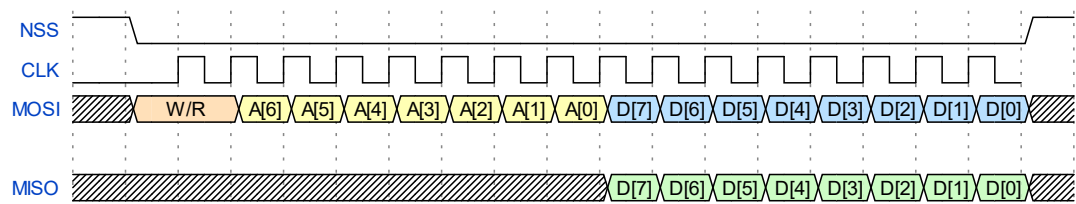


图 4-3 SPI 时序图（单次访问）

4.2.3 WiFi/4G 模块选型

针对智慧交通隔离带系统中网关连接 Internet 方式的不确定性，本网关在主板上预留一组扩展接口，可以在不同环境下的选择合适的网络模块，可采取的连接技术包括 WiFi、4G、5G 等。WiFi 模块适用于市区、有无线局域网的场景下，而 4G、5G 模块则更加适合在空旷的高速公路等非市区交通道路中使用。

针对使用 WiFi 访问 Internet 的场景，本文选择 AN1509 ATK-ESP8266 WiFi 模块（如图 4-4 所示），它支持 3.3V 和 5V 可选供电电压，本网关可直接对其进行供电。此 WiFi 模块与 MCU 通讯的接口为 USART，内置 AT 指令支持固件，故 MCU 端只需使用 STM32CubeMX 对 USART 外设进行简单的配置就可以直接使用 AT 指令配置 ESP8266 为 STATION、AP、AP+STATION 模式，极大减少了开发工作量。

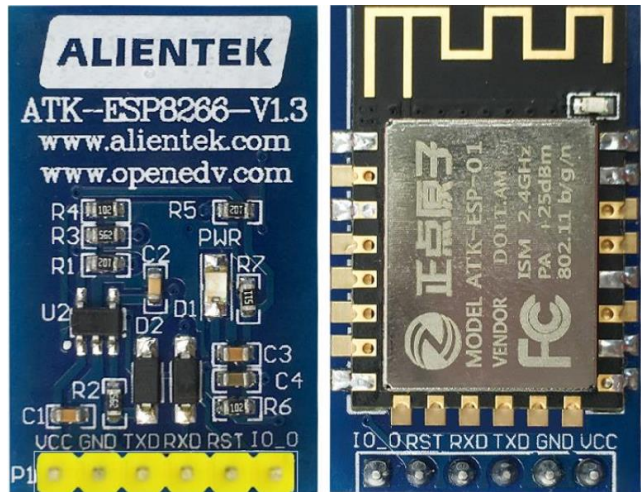


图 4-4 ATK-ESP8266 模块外观图

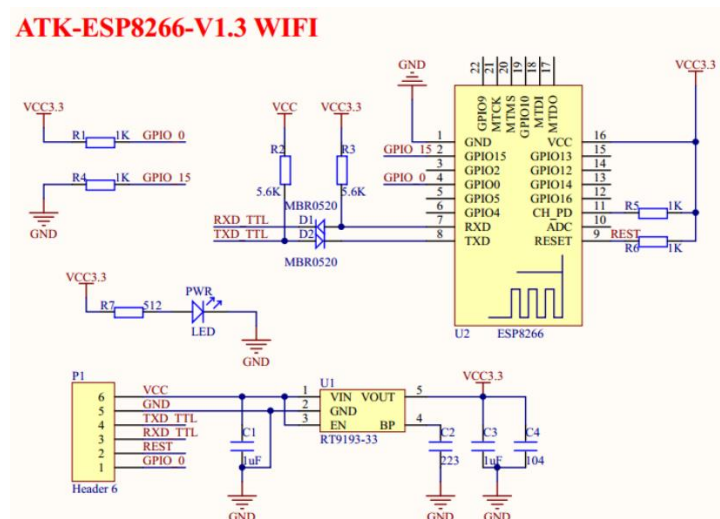


图 4-5 ATK-ESP8266 WIFI 模块原理图

4.2.4 显示模块选型

显示设备对于智慧交通隔离带系统的网关非常必要，终端连接情况、数据的上传和接收都可以用过显示设备直观的告诉管理员，同时，与按键相配合，还可以实现网关的本地配置，极大方便了系统维护人员进行安装调试和故障修复。

本网关选用 SSD1306 驱动的 0.96OLED 显示模块（如图 4-6 所示），OLED 显示器不需要发光背板，而是每一个像素都具有单独的光源，所以其具有对比度高（接近于 ∞ ）、耗电极低等优先。且由于没有发光背板所以通常可以做的比较薄，可以用于制作柔性面板，未来大有取代 LED 之势。

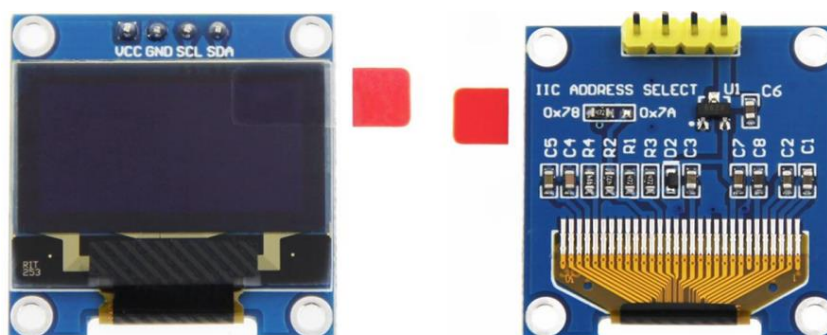


图 4-6 0.96 寸 OLED 显示模块实物图

SSD1306 支持 IIC、SPI 等多种通信方式，通过 BS0、BS1 和 BS2 对应引脚是否焊接电阻来修改通信方式，具体对应方式如图 4-7 所示：

SSD1306 Pin Name	I ² C Interface	6800-parallel interface (8 bit)	8080-parallel interface(8 bit)	4-wire Serial interface	3-wire Serial interface
BS0	0	0	0	0	1
BS1	1	0	1	0	0
BS2	0	1	1	0	0

图 4-7 SSD1306 通信接口的选择

此 OLED 模块使用 IIC 接口与 MCU 进行通讯，除供电线和 GND 之外，IIC 协议只需要一根时钟同步线和一根数据线就能完成主从设备的通信。IIC 的从设备地址选择需要模块电路上进行修改，如图 4-8 的红框内，通过电阻的焊接位置来设置 OLED 显示模块的从机地址，若两边同时焊接电阻，则相当于接了两个 OLED 屏。

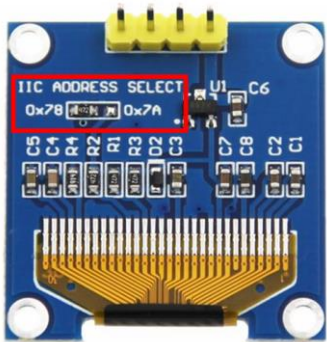


图 4-8 OLED 模块从机地址选择

4.2.5 串口调试模块选型

受应用场景限制，网关上的显示模块只能选择尺寸较小的显示器。有时，我们需要了解更多的网关信息或者进行更加复杂的交互，这时仅通过 0.96 寸的显示器和按键显然是无法完成的。所以本网关设计了一组串口调试的指令集，加之 MCU 端添加与指令解析相关的代码，如图 4-9 所示，PC 可以通过 USB 转串口模块与 MCU 进行通信，完成更加复杂的调试工作。

本网关选择 CH340 芯片实现 USB 转串口通信，它具有以下特点：

表 4-4 CH340 特点

封装形式	SSOP-20 和 SOP-16
供电电压范围	3.3V 或 5V DC
工作方式	全速 USB 设备接口，全双工通信
波特率范围	支持 2400bps 到 115200bps

兼容程度	可安装与 Windows、Linux、MacOS 等常见操作系统，与 CH431 驱动程序兼容
其它	可在外部添加简单的转换电路实现与 RS232、485 等串口兼容

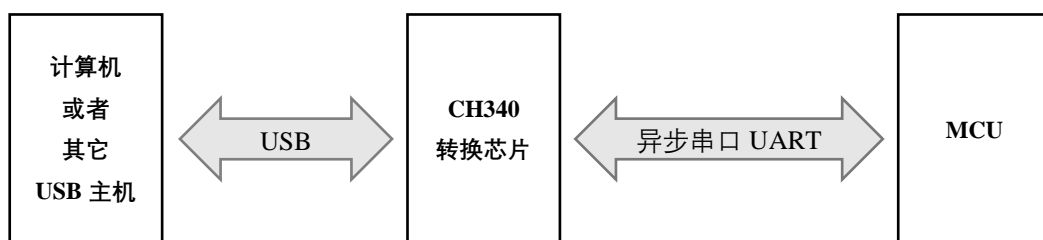


图 4-9 PC 与 MCU 通信示意图

4.2.6 电源模块选型

稳定的电源供电是系统正常工作的前提，在对网关供电的设计上要综合考虑每个模块的供电电压范围、最大工作电流、功耗等，如果供电电压与模块所需不匹配，则要选择合适的电压转换芯片满足每个模块的供电要求。

本网关的各个模块的供电要求如表 4-5 所示：

表 4-5 网关各模块供电要求

MCU	3.3V
LoRa 模块	2.7~3.6V
WiFi 模块	3.3V~5V
串口电平转换模块	5V
显示模块	3.3V

故对网关的电源进行设计时需要通过降压将 USB 输入的 5V 电压转换为 3.3V，本文采用 AMS1117-3.3 稳压芯片将 5V 电压降为 3.3V，同时满足不同模块对 3.3V 或 5V 的电压需求，是模块工作在最稳定的状态下。

AMS1117-3.3 稳压芯片的输入电压可以是常见的 5V、9V 和 12V（4.75~12V），如图 4-10 所示，本网关采用 USB 5V 输入，经过 DC-DC 转为 3.3V，广泛应用于单片机开发设计中。

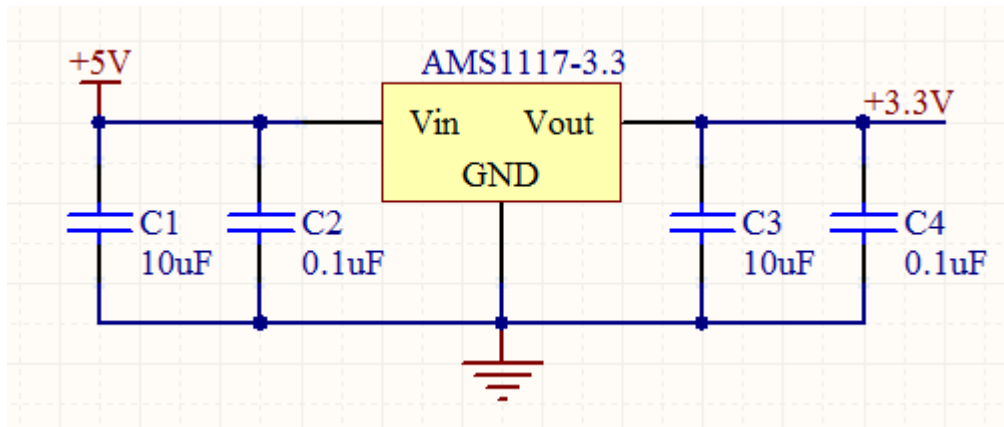


图 4-10 AMS1117-3.3 电路原理图

4.3 电路板布线及抗干扰设计

本网关的 PCB 采用两层板设计，具体要考虑的规则和技巧如下：

1. 电源和地的布线

- a) 将电源和 GND 平面层置于电路板内部，对于给各模块供电，建议使用多路电源进行供电，避免模块到模块之间的菊花链式的电源连接。
- b) 在电源、地线之间加上容值为 0.1uF 的去耦电容。
- c) 电源和 GND 的走线宽度要大于 10mil，具体走线宽的大小关系是： $GND > VCC > \text{信号线}$ 。
- d) 做正反两面 GND 网络的铺铜，在 PCB 上把没被用上的地方都与 GND 相连接作为地线用。
- e) WIFI 模块、LoRa 射频模块的天线下方禁止铺铜，防止信号对天线灵敏度产生干扰。

2. 信号线的布线

- a) 各元件之间要采用尽可能短的走线路径。
- b) 元件出线的地方先水平走线，一段距离之后再做角度，且最好避免使走线 90° 弯折。

3. 必要的散热设计

- a) 在元件的 Datasheet 中找到该软件的“热阻”等级，为等级高的元件增加散热设施（风扇、散热片等）。

- b) 对连接到 GND 铺铜的通孔或者过孔使用热风焊盘
- c) 关键元件尽量远离高热源。

4. 进行完整的 DRC 检查

4.4 本章小结

一套优秀的、出错率低的硬件是整个系统设计的关键，本章对 LoRa 网关的硬件设计做了详细的介绍，其采用模块化的设计方案，主要包括：MCU、LoRa 模块、WiFi/4G 模块、显示模块、串口调试模块和供电模块。

基于上一章充分的需求分析，从智慧交通隔离带的应用背景出发，对于每个模块做了元件的选型工作。对原理图、PCB 的设计布线等做出了规划，使整个网关的硬件系统处于良好的运行状态，为完成下一章节的软件设计提供坚实的基础。

5 网关系统软件的详细设计

众所周知，硬件是整个物联网系统设计的基础，那么网关的软件是整个物联网系统设计的核心，负责基于设计好的硬件实现最终的逻辑功能。网关系统的软件设计从上至下包括应用层、中间件层、操作系统层、板级支持包和驱动层五个层次。本章将对 LoRa 网关的软件设计进行详细的介绍。

5.1 网关系统的软件框架

网关系统软件框架设计如图 5-1 所示，软件设计大致可分为五层来实现，分别是应用层、中间件层、操作系统层、板级支持包和驱动层，按照高内聚、低耦合的思想，上层能调用下层的函数接口，而下层函数不知道上层函数的存在，每层之间相互独立。

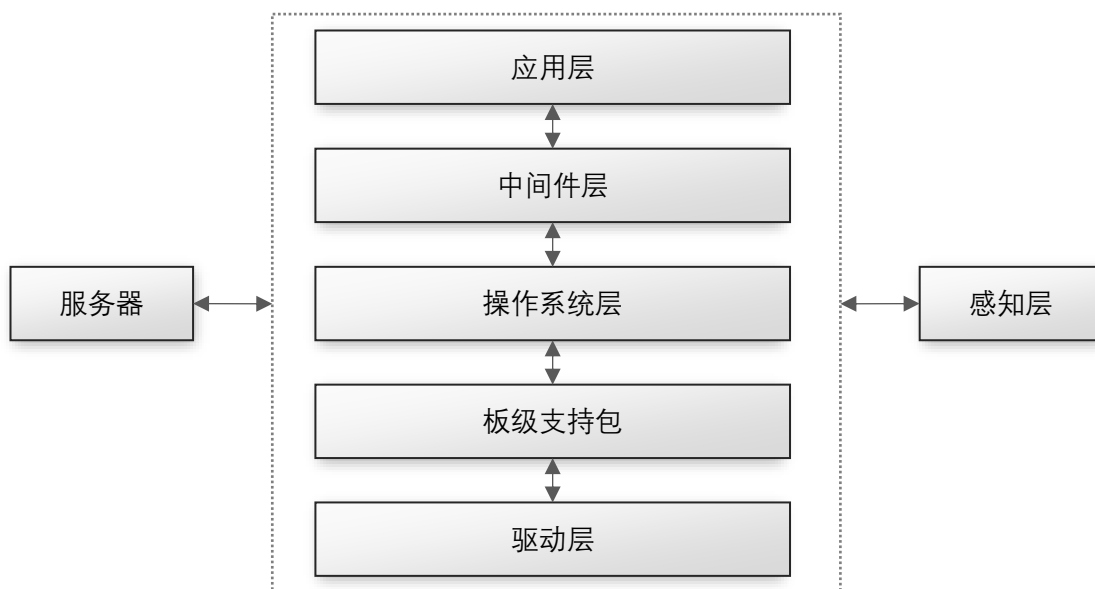


图 5-1 网关软件框架设计图

5.2 驱动层

驱动层是实现对 STM32 内部各类外设的配置以及基础功能函数的封装，并向上层提供接口，方便其利用 STM32 的外设完成硬件要实现的功能。

相较于直接对繁多复杂的寄存器进行读写，ST 公司为开发者提供了三种不同驱动库文件，分别是标准外设库(STD 库)、HAL 库和 LL 库。其中本网关采用的 HAL 库是 ST 公司为 STM32 的 MCU 最新推出的嵌入式软件库，比标准库的代码整合度、抽象度更高，HAL

库致力于定义一套通用的用户友好的外设 API 函数接口，从而可以极大提高程序的可读性和可移植性。

HAL 库目前已覆盖绝大多数的 STM32 产品，ST 公司推出的可视化配置工具 STM32CubeMX 也仅支持 HAL 库和 LL 库，所以网关的驱动层选择 HAL 库也会极大减少开发工作量。




Offer		Portability	Optimization (Memory & MIPS)	Ease of use	Readiness	Hardware coverage
 STM32Snippets			+++			+
 Standard peripheral library		++	++	+	++	+++
 STM32 Cube	HAL APIs	+++	+	++	+++	+++
	LL APIs		+++			++

图 5-2 STM32 各类库的对比

5.3 板级支持包

板级支持包（Board-level Support Package）是“硬件模块的驱动层”，它实现了对各类外部硬件模块如 LoRa 模块、WiFi 模块、OLED 模块等的初始化和基本读写操作函数的封装，屏蔽了复杂的底层逻辑，向上层提供统一的接口，实现了软件模块内部高内聚、模块间的低耦合。板级支持包需要调用硬件驱动接口并且向操作系统层提供接口，其主要作用包括：

1. 初始化硬件设备模块，主要是 LoRa 模块、WiFi/4G 模块、显示模块。
2. 完成对硬件设备基础操作的函数封装，例如存储字节、发送字节、显示字符等。
3. 检测系统硬件是否正常。
4. 完成操作系统启动前的准备工作并引导操作系统的启动。

5.3.1 LoRa 模块初始化

1. LoRa 模块与 MCU 通过 SPI1 进行通讯，故 LoRa 模块的初始化首先要对 STM32 的 SPI1 进行初始化操作，并且按照硬件连接初始化 LoRa 模块的片选引脚。

2. 对 LoRa 模块进行硬件复位。
3. 通过读取 SX1278 的 RegVersion(0x42) 寄存器，比较值是否等于 0x12（芯片手册中给出）来判断 MCU 与 LoRa 模块之间是否成功的建立通讯连接。
4. 执行 LF 和 HF 波段的接收链校准，此操作必须在芯片刚复位完成后马上执行，此时的寄存器都应当位于默认值状态。
5. 接收完成、发送完成等事件的通知是通过引脚 DIO0~DIO5 产生不同信号来实现的，故要对与这些引脚连接的 MCU 引脚进行初始化和中断配置。
6. 此芯片只有当处于睡眠模式时才能对寄存器进行读写，所以在对寄存器进行修改之前，要先将芯片设为睡眠状态，修改完成后，恢复 LoRa 模式。

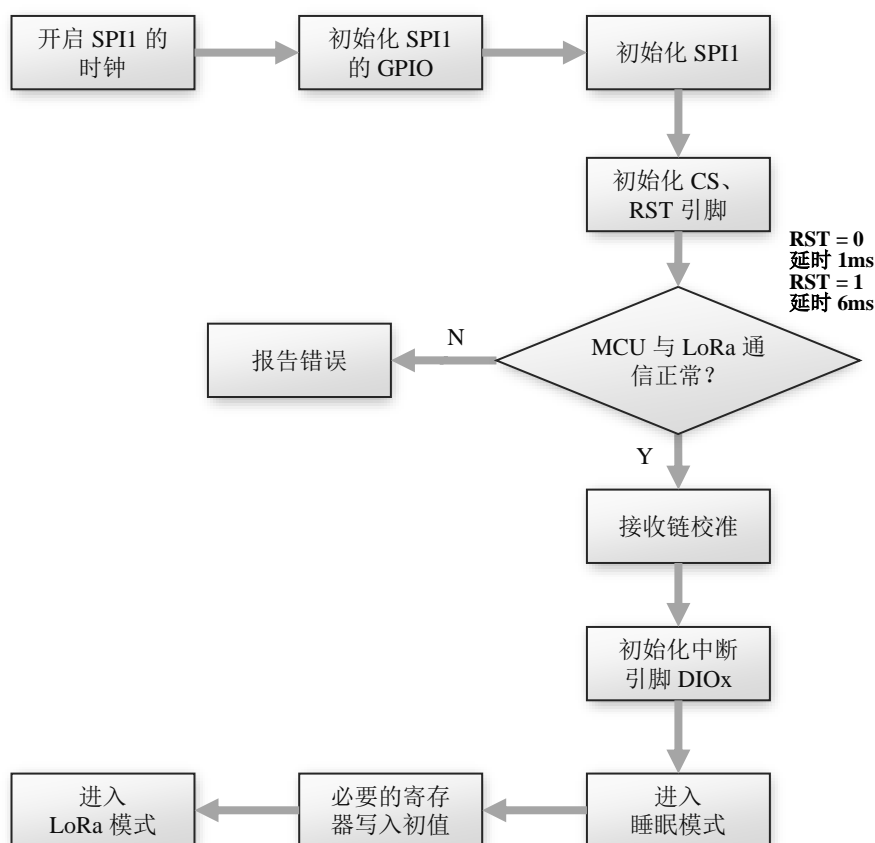


图 5-3 LoRa 模块初始化框图

5.3.2 WiFi 模块初始化

1. WiFi 模块与 MCU 通过 USART 进行通讯，故先要使用 CubeMX 工具初始化 USART 外设。

2. 由于 WiFi 模块内部集成的 ESP8266 已刷入支持 AT 指令的固件，可以直接向它发送“AT”命令来检测 WiFi 模块与 MCU 是否成功连接，若 WiFi 模块应答“OK”，表示连接成功。
3. WiFi 模块连接无误后，就要对其进行配置。由于本网关中 WiFi 模块的作用是连接 Internet，所以应将其配置为 STA 模式，并输入正确的 SSID 和 PASSWORD 以连接互联网。
4. WiFi 模块的配置完成后，要开启透传模式，以便与服务器建立 MQTT 连接（这部分将在中间件层完成）。

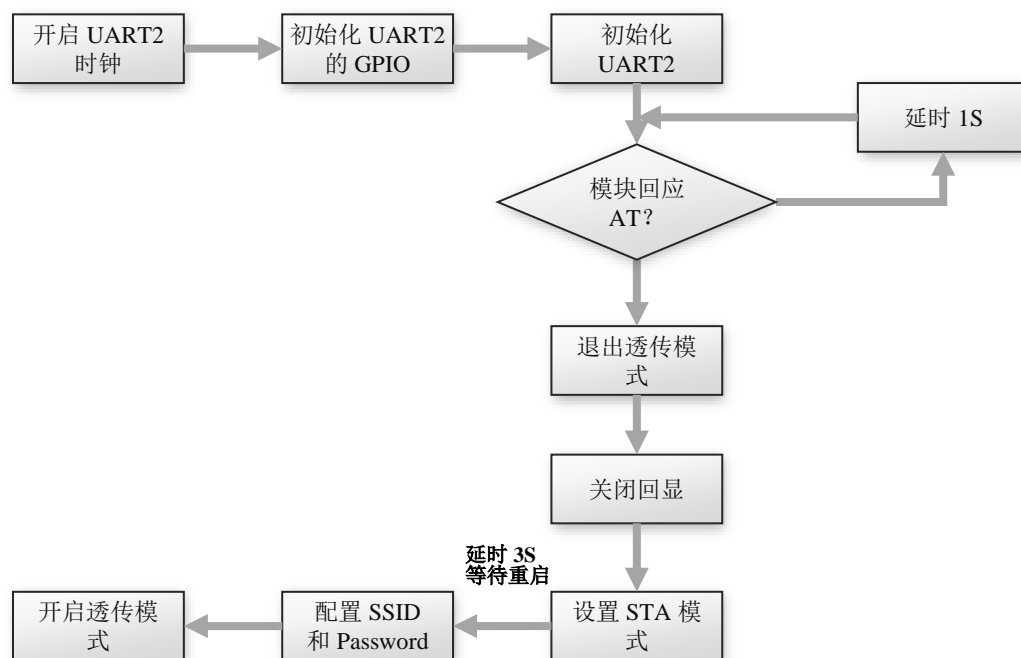


图 5-4 WiFi 模块初始化框图

5.3.3 显示模块初始化

1. OLED 显示模块通过 IIC 接口与 MCU 通讯，故先要通过 STM32CubeMX 对 IIC 外设进行初始化配置。
2. 显式模块使用的驱动 IC 为 SSD1306，IIC 通信接口初始化完成后，根据 SSD1306 DataSheet 的说明，要先对 SSD1306 进行复位，即将 RES# 引脚进行置 0，延时 10ms 后再置 1；
3. 驱动 IC SSD1306 通过 IIC 发送命令对其内部的寄存器进行配置。
4. 显存就是我们定义在 MCU Flash 中的一块空间，我们将其定义为 128*64 字节的

二维数组。显式模块的像素个数也是 128*64 个，和显存的元素一一对应，将对应位置 1 表示点亮该像素点。所以在初始化时，我们要将显存全部清零。

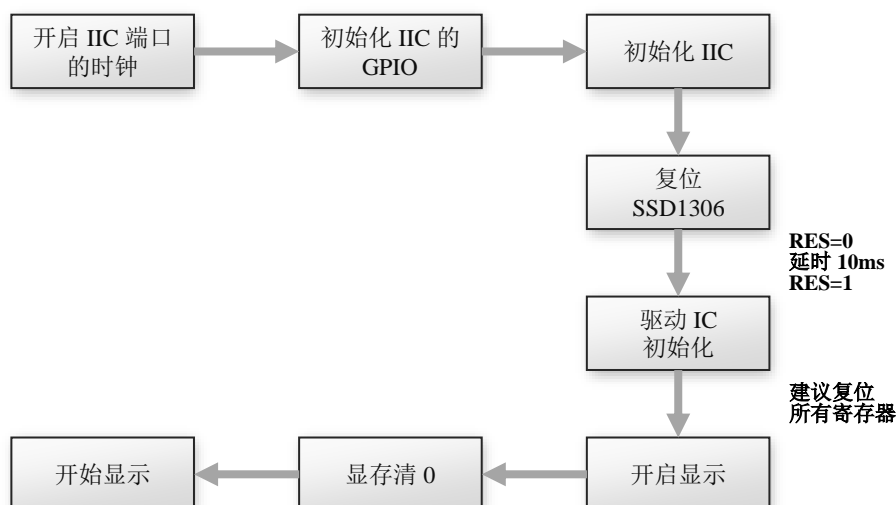


图 5-5 显示模块初始化框图

5.4 操作系统层

由于智慧交通隔离带的应用场景下对数据的传输具有一定的实时性要求，本网关移植了 FreeRTOS，该系统体积小、开源、免费，且移植非常方便，使用 STM32CubeMX 就可通过图形化配置方法来移植 FreeRTOS 到项目工程中。

5.4.1 STM32CubeMX 配置 FreeRTOS

在 STM32CubeMX 打开的项目工程中，左侧选择 FreeRTOS 选项（如错误!未找到引用源。所示），就能在此项目中移植 FreeRTOS 并对其进行参数配置。

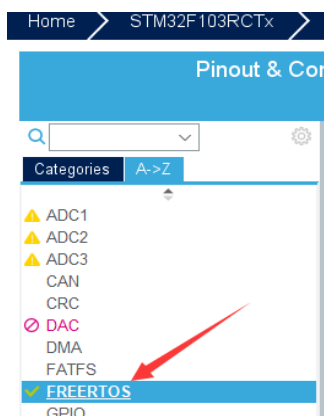


图 5-6 STM32CubeMX 中 FreeRTOS 选项

FreeRTOS 的配置步骤如下：

1. 选择 FreeRTOS 中间层模式

CMSIS_v1 为更多的 RTOS 提供支持，而 CMSIS_v2 相比 CMSIS_v1 兼容更多的 CPU 架构和实时操作系统。本项目中我们使用 CMSIS_V1。

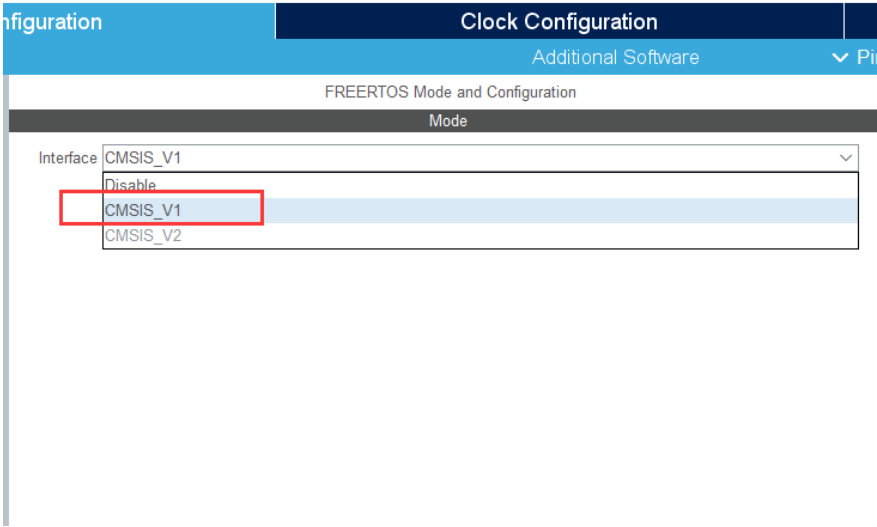


图 5-7 FreeRTOS 中间层配置

2. 配置 FreeRTOS

在 Config parameters 选项卡中我们可以根据需求对 FreeRTOS 的各种功能进行配置和裁剪。之后应用层进行任务的创建也是位于 STM32CubeMX 的 FreeRTOS 分支中的 Task and Queues 选项卡中进行操作，软件定时器、信号量等常用的功能也可以在 STM32CubeMX 中方便的进行配置。

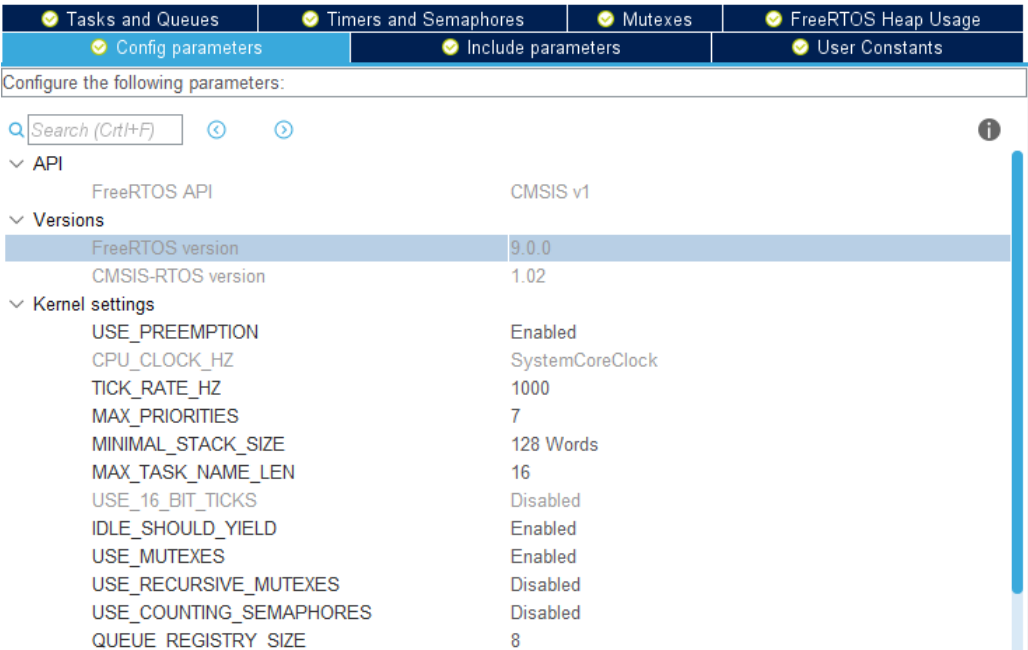


图 5-8 FreeRTOS 参数配置

5.4.2 初始化硬件模块

操作系统层的另一个重要功能是调用板级支持包提供的接口对各个硬件模块进行初始化，使整个系统在进入任务调度之前全部初始化完成，防止产生非法操作。另外，各个模块的初始化过程前后应当使用互斥/二值信号量进行保护，保证初始化过程不会被中断而产生错误。

5.5 中间件层

中间件层负责实现一系列综合性的功能函数，每个函数可能调用多于一个的板级支持包函数接口和操作系统接口，并向上提供接口便于应用层直接调用，另外，中断服务函数的实现也是在中间件层完成。本节将列举几个重要的功能函数。

5.5.1 连接 MQTT 服务端

若客户端想与服务端建立 MQTT 连接，必须主动发送连接报文并等待服务端回应连接 ACK 报文，此报文需要配置以下的参数：

表 5-1 MQTT 请求连接报文参数

协议名（必须是 MQTT）
MQTT 协议版本号
客户端 ID
清除会话标志
遗嘱标志
遗嘱 QoS
用户名
密码

TCP 连接建立后，如果服务端没有在合理的时间内收到客户端发来的请求连接报文，服务端应断开此 TCP 连接。在接收到客户端的连接请求报文后，服务端需要判断其是否符合规范，如果报文不符合规范，服务端将直接关闭 TCP 连接。服务端也可以对连接请求报文的内容中的身份和授权进行验证和检查。

TCP 的连接不会一直保持，在 MQTT Dashboard 中可以配置若 20S 服务端没有收到此客户端发来的连接请求报文，那么服务端会放弃此连接。当然，服务端即使收到了连接请

求报文，也要检查报文携带的信息是否符合规范，如果此连接报文不符合任一规范，服务端应该发送一个响应码非零的请求回复报文，并且关闭此 TCP 连接。

如果验证成功，服务端会执行下列步骤：

1. 将连接报文中有效载荷中客户端标识符与已连接的客户端的标识符进行比对，若发现相同的另一个客户端存在，那么就必须断开之前的连接，启动新的 MQTT 连接。
2. 服务端执行清理会话的过程。
3. 服务端必须发送请求回复报文对客户端的连接进行确认，其返回码必须为零。
4. 开始消息分发和保持连接状态监视。

客户端在发送连接请求报文后需要等待服务端返回请求回复报文，并判断其中的连接返回码是否为 0x00（表示连接已被服务端接收），若不是，则按照对应的标识做出错处理。

表 5-2 CONNACK 报文连接返回码的值

值	返回码响应	描述
0	0x00 接收此连接	连接已被服务端接受
1	0x01 拒绝此连接，协议版本无效	此协议级别不被支持
2	0x02 拒绝此连接，客户端标识符无效	服务端不允许使用此客户端标识符
3	0x03 拒绝此连接，服务端出错	MQTT 服务端当前状态不可用
4	0x04 拒绝此连接，用户名、密码出错	用户名和密码与数据库不匹配
5	0x05 拒绝此连接，无授权	不被授权的连接请求
6-255	保留	

5.5.2 MQTT 的订阅和发布

MQTT 协议基于订阅/发布的模式来发送和接收数据的，订阅者和发布者之间通过 TOPIC（主题）来实现定向传输消息。作为智慧交通隔离带的网关，它通过 MQTT 协议向上发布传感数据，而服务器应用同样作为 MQTT 客户端订阅与发布操作相同的主题（可使用通配符），MQTT 服务端就能将消息正确转发给服务器应用。

1. 订阅消息

客户端通过向服务端发送 SUBSCRIBE 报文来订阅 TOPIC，也可以使用通配符来一次订阅多个 TOPIC，服务端若不支持使用通配符则应当对此 SUBSCRIBE 报文返回

特定响应码来拒绝该请求。

当客户端收到服务端发来的订阅确认报文后，服务端就会将此 TOPIC 下的消息转发给该客户端，MQTT 服务端支持多个客户端订阅同一个 TOPIC。

2. 发布消息

客户端通过 PUBLISH 报文来发布消息，此消息会指定一个 TOPIC（也支持使用通配符）指定多个，客户端的消息会先发送到 MQTT 服务端，服务端负责筛选相同的 TOPIC 进行消息转发。

对于多个客户端订阅同一 TOPIC 的情况，服务端转发消息的顺序应当和 Qos 等级顺序保持一致。

5.5.3 LoRa 的发送和接收

在发送/接收数据前，要配置 LoRa 模块的发送/接收参数，主要包括：

表 5-3 LoRa 发送/接收参数

调制模式	发送和接收
发射功率	仅发送
扩频因子	发送和接收
纠错编码率	发送和接收
前导码长度	发送和接收
报头模式	发送和接收
负载长度	发送和接收
CRC 校验	发送和接收
跳频开启/关闭	发送和接收
跳频周期	发送和接收
接收模式	发送和接收
发送超时时间	仅发送

1. 发送数据

发送模式下，MCU 将数据用户负载 (payload) 发送到 SX1278 的 FIFO，SX1278 的数据包处理器在发送之前将对数据负载拼接上前导码和同步字，若开启 CRC 功能则还要进行 CRC 校验，并在末尾加上 2 字节的校验和。

2. 接收数据

接收模式下，等待 DIO0 传来的中断信号，在中断服务函数中对接收到的 LoRa 数据包进行解析以得到用户负载字段，解析过程主要包括：删除前导码和同步字字段，若启动 CRC 功能则进行 CRC 校验，检查数据包携带的校验和是否正确，若正确则将数据包负载发送到 SX1278 的 FIFO，FIFO 中只含有每个数据包的负载段。

5.5.4 解析 JSON 数据

服务器下发到网关的数据都是 JSON 格式的字符串，在网关中需要将它进行分离以得到我们需要的各类参数数据，再向对应的传感结点发送命令。本网关引入了 jasson 库来帮助实现上述过程。

JSON 数据解析的过程如下所示：

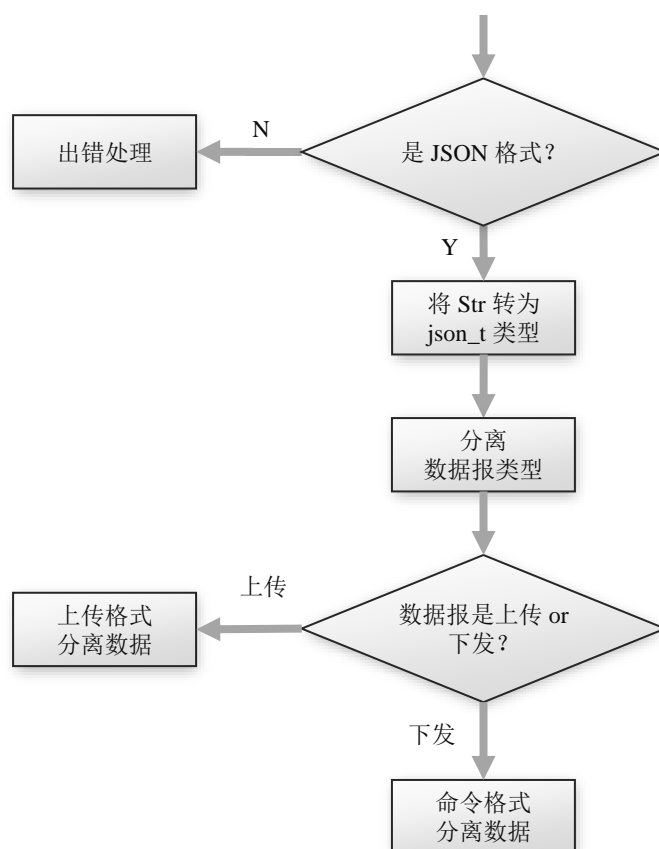


图 5-9 解析 JSON 数据流程图

5.5.5 中断服务函数实现

中断服务函数的定义是在板级支持包实现的，而其实现则位于中间件层。本网关系统中主要包含的中断服务函数有：

1. USART1 的接收中断；USART1 与 PC 连接作为调试接口使用，USART1 的中断处理能够保证正确接收来自 PC 的串行数据。
2. USART2 的接收中断；USART2 与 WiFi 模块连接，向 ESP8266 发送 AT 指令的响应、TCP 和 MQTT 协议的连接过程都需要 USART2 接收中断的参与。
3. TIMER3 溢出中断；MQTT 是基于 TCP 的协议，需要定时发送心跳包来保持连接，TIMER3 是 STM32 外设中的高级定时器，它的中断服务函数用于定时发送心跳包。
4. EXTIx 外部中断；LoRa 模块的 DIO0~DIO5 引脚与 MCU 连接，产生特定信号来通知 MCU 某些操作的完成或出错等事件，所以 MCU 对应引脚的中断服务函数用于对这些事件做出相应的处理动作。

5.6 应用层

应用层通过采用合适的调度策略，调用中间件层向上提供的接口，实现基于智慧交通隔离带场景下的应用。由于本系统移植了 FreeRTOS，所以整个应用层的事件调度由 FreeRTOS 内核的任务调度器实现，开发者需要创建和配置任务，包括 MQTT 数据收发任务、LoRa 数据收发任务、OLED 显示任务，使用事件标志组等功能来完成应用层逻辑。

5.6.1 MQTT 数据收发任务

网关与服务器之间基于 MQTT 协议进行数据传递，此任务的业务流程如所示：

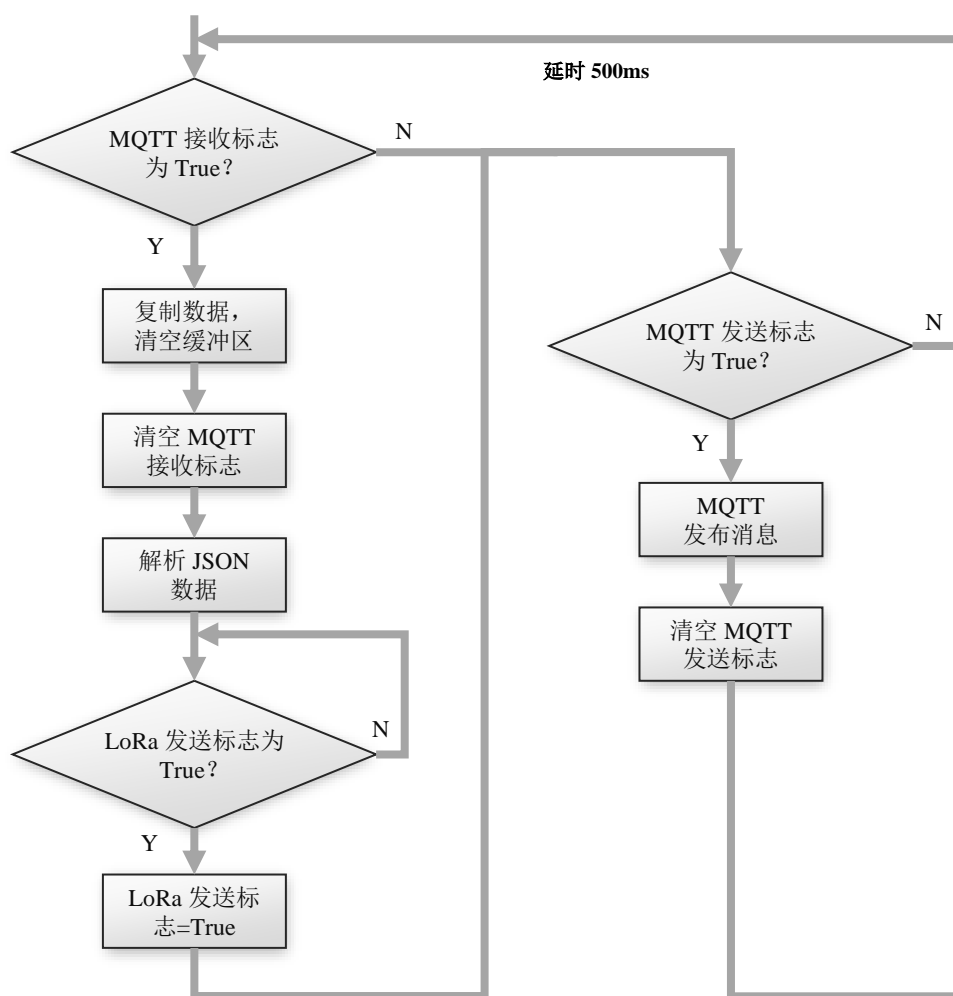


图 5-10 MQTT 数据收发任务流程图

5.6.2 LoRa 数据收发任务

网关与无线传感结点之间通过 LoRa 协议进行数据传输，在 FreeRTOS 中创建 LoRa 数据收发的任务用于处理发送和接收 LoRa 数据的业务，具体的流程如所示：

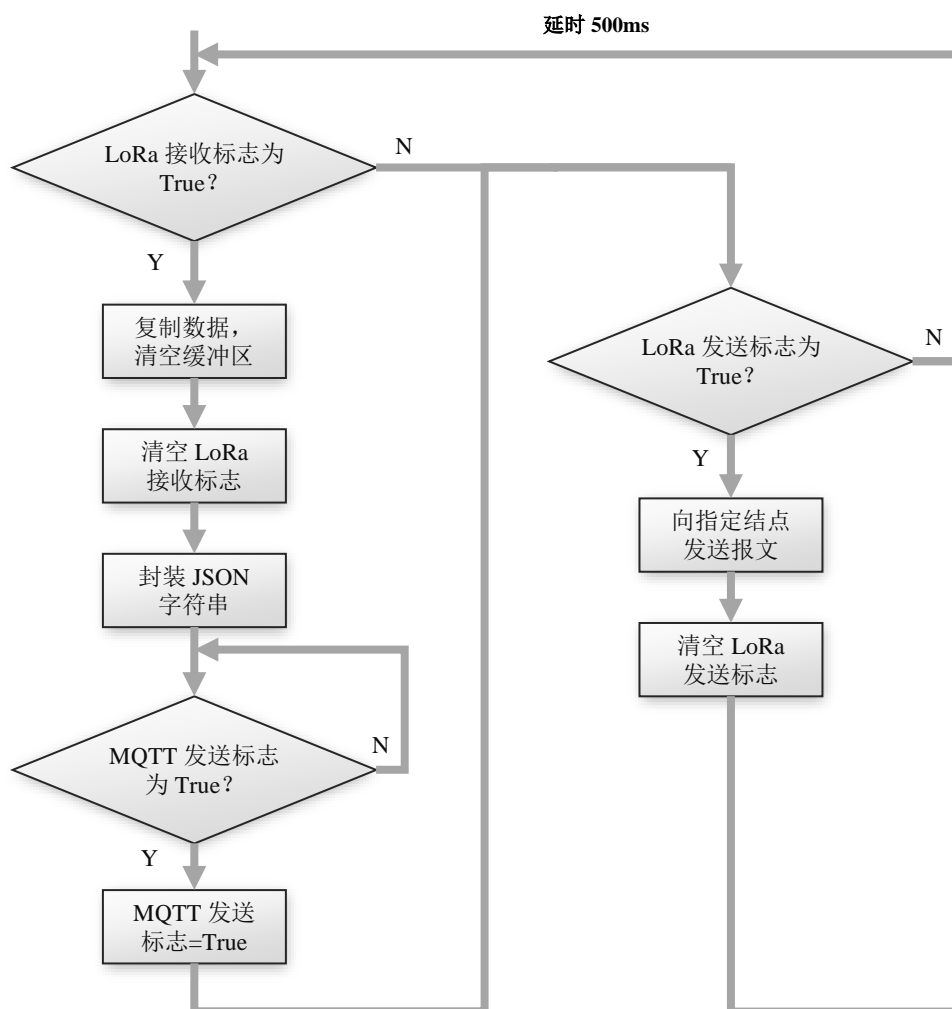


图 5-11 LoRa 数据收发任务流程图

5.6.3 OLED 显示任务

OLED 显示任务用于显示系统当前的事件和状态，主要包括 WiFi 连接状态、MQTT 连接状态、MQTT 发送数据事件、LoRa 发送数据事件，其主要作用为调试使用，也可增加其它状态的显示。其主要任务流程如图 5-12 所示：

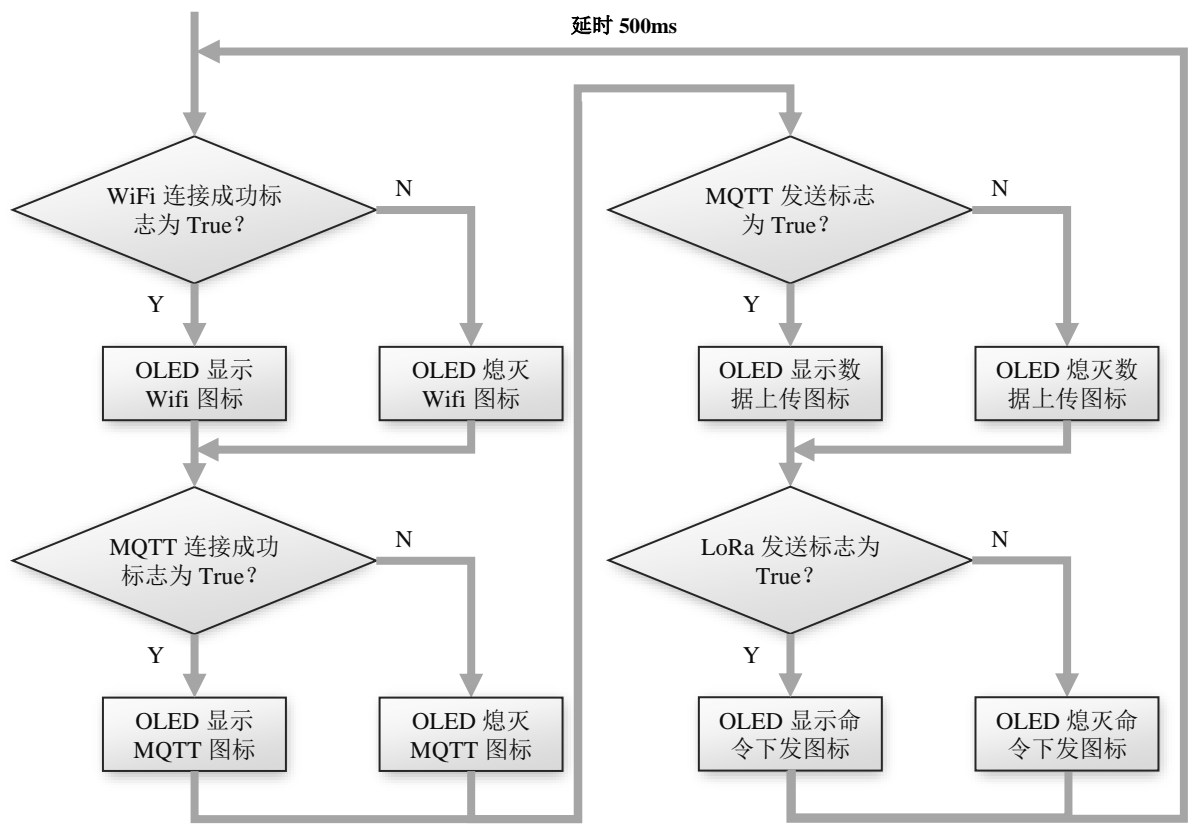


图 5-12 OLED 显示调试任务流程图

5.6.4 事件标志组

本文基于三个任务来实现数据转发、协议转换和显示调试，但实际上每个任务并不是完全的相互独立，有时候任务之间也需要传递某些信息。比如，在 MQTT 连接服务器成功之后，希望在 OLED 屏幕上显示 MQTT 已连接字段；网关收到来自终端结点上传的数据，需要使用 MQTT 协议向上转发等。

通过使用 FreeRTOS 提供的事件标志组就能完成上述的需求，针对本网关的情况，现使用其中 7bit 的标志，具体分布情况如表 5-4 所示：

表 5-4 网关使用的事件标志

位	事件标志位名称	事件标志位含义
bit0	EVENTBIT_WIFI_CONNECTED	WiFi 连接成功
bit1	EVENTBIT_MQTT_CONNECTED	MQTT 连接成功
bit2	EVENTBIT_MQTT_RXMSG	MQTT 收到消息
bit3	EVENTBIT_MQTT_TXMSG	MQTT 需要发送消息
bit4	EVENTBIT_MQTT_TXHB	MQTT 需要发送心跳
bit5	EVENTBIT_LORA_RXMSG	LoRa 收到消息
bit6	EVENTBIT_LORA_TXMSG	LoRa 需要发送消息
bit7	EVENTBIT_OLED_UPLOAD	OLED 显示上行传输
bit8	EVENTBIT_OLED_DOWNLOAD	OLED 显示下行传输

在事件发生的时候，将其事件标志位置 1，在需要对事件处理的位置轮询检查该事件标志位，处理完成后还要将该事件标志位清 0。

6 总结与展望

本章将总结本篇论文的主要研究内容，并展望 LoRa 网关未来的发展。

6.1 总结

截至目前已有近 20 个国家公布了全面引入 LoRa 网络的己画，然而国内 LoRa 网络的建设没有得到充分的重视。本论文的目的在于设计一种应用于智慧交通隔离带背景下的 LoRa 网关，实现高效率、安全性强的协议转换、数据转发等功能，让物联网技术更好的渗入智慧交通系统管理体系、为人们提供更加高效、智能的交通出行环境。

参照智慧交通隔离带网关系统应具备的功能需求，本网关的设计从硬件和软件两方面入手。其中，硬件设计采用模块化的设计思想，各模块之间相互独立又彼此连接，网关的软件设计按照分层的方法，基于 FreeRTOS 嵌入式实时系统进行开发。具体工作总结如下：

1. 对网关的功能需求进行了详细的分析，从功能性和安全性、实时性等多个方面进行需求讨论，整体上确定了网关系统所需要达到的性能目标。
2. 参照第三章做出的需求分析，对网关硬件模块进行选型和主板 PCB 设计，能够交通道路应用场景下实现稳定工作。
3. 参照第三章做出的需求分析，分层次对网关进行软件开发，实现协议转换、数据转发、接入管理等功能，将项目部署于 FreeRTOS 系统。

通过对网关硬件的测试，各个模块工作电压、电流均处于其正常工作范围，电路中没有发烫过热的现象。软件方面，通过 FreeRTOS 系统中的运行时间 API 函数测得处理单个 LoRa 数据包的时间小于 2ms，处理单个 MQTT 报文的时间小于 5ms，其处理数据的效率符合现实场景下的需求，

本文还对 LoRa 传输中丢包率进行了测试，在接入 20 个终端结点的情况下，每个结点共向网关发送 100 条数据，最终 LoRa 网关共接收到正确消息 1993 条，丢包率小于 0.5%，具有较高的可靠性。

6.2 展望

本文从智能交通隔离带系统的实际需求出发，从硬件和软件方面设计实现了 LoRa 网关。虽然已对基本功能进行了实现，但整个网关仍存在以下几个方面的不足，需要在将来

的设计中再次完善设计。

1. 网关与终端结点之间的传输是明文传输，未进行加密处理。LoRa 数据包一旦被截获其内容就暴露于他人或组织，对车辆及行人的生命安全造成威胁。故在之后的设计中要首先考虑将 LoRa 数据包进行加密处理，或采用其它方式提供更安全的数据传输方式。
2. 对通信环境的要求比较苛刻，网关使用 LoRa 模块与上层通信，而真实情况中很难实现全道路的无线局域网铺设，故在之后的设计中要根据所安装地点的实际环境，设计可选择的 WiFi/4G/5G 通信模块，仅需要对硬件模块进行更换，无需修改软件中的代码。
3. 出错处理的设计没有考虑足够多的情况。由于该网关布局在室外，现实状况下，尤其是碰到恶劣的天气时，网关可能出现需要无法预知的错误。为了确保整个系统的稳定运行，对这些无法预知的错误，就要设立尽可能完整的出错处理机制。未来通过完整的系统测试阶段，需要记录各类错误发生的环境、表现等，设立针对性的处理机制，使整个智能交通隔离带系统更加完善。

致谢

行文至此，代表着我在青岛大学的四年本科求知生涯结束，即将迈入人生的下一个阶段。在此，我想对我的母校、家人、老师和同学表达我来自内心深处的感谢。

首先我要感谢我的母校——青岛大学，这所坐落在青岛东南沿海的学府承载了我四年的美好回忆，“明德，博学，守正，出奇”的校训激励着我不断汲取知识、开拓眼界。

其次我要感谢我的父母，因生活所迫，他们早早放弃学业，将双手奉献给脚下的土地和清晨的雨露。正是由于他们的辛勤付出才能让我心无旁骛地投入学海之中。

需要特别感谢的是范延滨老师，也就是本篇论文的指导教师。从大二时对物联网的懵懂无知，到今天完成智能物联网网关的设计与实现，我向前迈的每一步都有范老师的支持与陪伴。我还清楚地记得范老师带我参加过的比赛、做过的项目，也忘不了他耐心解答我的每一个幼稚的疑惑。在我心里，范老师既是良师，也是益友，大四考研的那段时间，范老师也给了我很大的帮助和支持。

最后不能忘了感谢我朝夕相处的同学们，尤其是 409 寝室的各位兄弟，在我遇到学习、生活上的困难时总能给我帮助。我永远忘不了大家一起上课、学习的日子，也忘不了篮球场上大家挥洒汗水、气喘吁吁的模样，若没有你们我的大学生活将缺少许多色彩。

逝者如斯夫，不舍昼夜。不知不觉就要和很多人说再见了，大家都奔赴世界各地为自己而打拼，此次分别也不知下次何时才能重逢，希望我们都能实现心中最初的梦想，也请不要忘记在青岛大学的四年美好时光。

参考文献

- [1] 曾泽熠. 基于 CoAP 的家庭网络通信协议的设计与实现[D]. 北京:北京交通大学,2013. DOI:10.7666/d.Y2428212.
- [2] 温家宝, 2010 年政府工作报告, 新华社, 北京: 2010. http://www.gov.cn/20101h/content_1555767.htm. 2010 年 3 月 15 日
- [3] 程春,刘洋,李伟亭. 物联网技术在水资源监测中的应用[J]. 数字技术与应用,2017(1):74-75. DOI:10.3969/j.issn.1007-9416.2017.01.045.
- [4] 李洁. 浅谈智慧交通在智慧城市建设中的作用[J]. 中国宽带,2021(3):73.
- [5] 张若雯,冀强,崔媛媛,石佳.基于智能交通潮汐车道中可移动隔离带的设计与实现[J].电脑知识与技术,2018,14(28):286-288.
- [6] 黄明浩. 一种安全的物联网网关设备软件设计与实现[D].电子科技大学,2017.
- [7] 庄旭菲,刘志强,高睿鹏.基于嵌入式 Linux 的农业物联网网关软件的研究与设计[J].信息与电脑(理论版),2019,31(20):77-78.
- [8] 赵佳奇. 基于 LoRa 技术的智慧消防系统研究[D].吉林大学,2020.
- [9] 徐侃,丁强.一种基于 MQTT 协议的物联网通信网关[J].仪表技术,2019(01):1-4+43.
- [10] 徐卸土. LoRa 移动物联网基站无线覆盖分析[J]. 电信技术,2017(8):38-40. DOI:10.3969/j.issn.1000-1247.2017.08.010.
- [11] 王柯雅. 基于位置的企业访客管理平台[D]. 江苏:苏州大学,2018.
- [12] Yokotani T, Sasaki Y. Transfer protocols of tiny data blocks in IoT and their performance evaluation[C]. 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). IEEE, 2016: 54-57
- [13] Yassein M B, Shatnawi M Q, Aljwarneh S, et al. Internet of Things: Survey and open issues of MQTT protocol[C]. 2017 International Conference on Engineering & MIS (ICEMIS). IEEE, 2017: 1-6
- [14] Andrew Banks, Ed Briggs, Ken Borgendale. MQTT Version 5.0[EB/OL]. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>
- [15] 吴俊辉, 吴桂初, 陈冲, 等. 基于 MQTT 协议的物联网网关设计[N]. 温州大学学报(自然科学版), 2019, 40(4): 54-61